



***MultiGen-Paradigm***  
VISUALIZE REALITY

**MultiGen-Paradigm, A Computer Associates Company,  
Distributed Interactive Simulation Tutorial**

MultiGen-Paradigm Inc. PROVIDES THIS MATERIAL AS IS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MultiGen-Paradigm, A Computer Associates Company may make improvements and changes to the product described in this manual at any time without notice. MultiGen-Paradigm, A Computer Associates Company, assumes no responsibility for the use of the product or this manual except as expressly set forth in the applicable MultiGen-Paradigm, A Computer Associates Company, agreement or agreements and subject to terms and conditions set forth therein and applicable MultiGen-Paradigm Inc. policies and procedures. This manual may contain technical inaccuracies or typographical errors. Periodic changes are made to the information contained herein: These changes will be incorporated in new editions of the manual.

Copyright ©2001 by MultiGen-Paradigm, A Computer Associates Company, World rights reserved.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, or other record, without the prior written permission of MultiGen-Paradigm, A Computer Associates Company. Duplicate this document without permission and you will be shot like a rabid dog in the street.

Use, duplication, or disclosure by the government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause and DFARS 52.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

LynX™ and Vega™ are registered trademarks of MultiGen-Paradigm Inc.

AudioWorks™ Dismounted Infantry Guy (DI-Guy)™ Large Area Database Manager (LADBM)™ Light Lobes™ Marine™ Navigational and Signal lighting (NSL)™ Non Linear Distortion Correction (NLDC)™, SensorVision™ Special Effects (FX)™ Symbology™ Terrain Fade Level of Detail (TFLOD)™ and Vega Vega Class Recorder (VCR)™ are registered trademarks of MultiGen-Paradigm Inc. MultiGen™, OpenFlight™, and Flight Format™ are registered trademarks of MultiGen-Paradigm Inc.

DfadGen, GameGen, GameGenII, OpenGame, SmartScene, SmartPalette, Take Flight and the MultiGen logo are trademarks of MultiGen-Paradigm Inc.

MultiGen II, MultiGen Creator are trademarks of MultiGen-Paradigm Inc.

Silicon Graphics™, IRIS™, and the Silicon Graphics logo are registered trademarks of Silicon Graphics, Inc. Adobe Photoshop™ is a trademark of Adobe Systems, Inc., which may be registered in certain jurisdictions MS-DOS™ is a registered trademark and Windows NT™ is a trademark of Microsoft Corporation in the United States and/or other countries.

UNIX™ is a trademark of AT&T Bell Laboratories.

All other trademarks herein belong to their respective owners.

Printed in the U.S.A.

## Prerequisites:

Successful completion of LynX Tutorial.

## Introduction:

This exercise will demonstrate all the necessary steps to create a simple DIS application to send and receive Protocol Data Units (PDUs) over your network.

The basic steps are 1) create your objects 2) configure DIS attributes for each object:

The main DIS components are :Inbound Template Entity Information, Outbound Entity Information and Outbound Entity Appearance.

*Inbound Template Entity Information* is used to define the force and type for the tank which will be used to map a specific tank to this template object. When the object is determined to be the best match, it will be copied and added to the DIS-HLA scene automatically.

*Outbound Entity Information* dialog contains information regarding the outbound network entity which may be associated with any Vega object.

*Outbound Entity Appearance* determines the initial state of the outbound entity when it is broadcast.

Munitions mapping, appearance mapping, and articulation mapping features are also supported by DIS, however those subjects are out of the scope of this tutorial.

Please see the example code and the Vega Options Guide for more details regarding these features.

## Tutorial:

### Step 1: Invoke LynX

NT

Start → Program Files → Vega → LynX

IRIX

Open a winterm

Type 'lynx' on the command line.

### Step 2: Turn off unnecessary modules

Before you begin creating your DIS application, let's turn off all add-on modules that will not be used for this exercise.

Under the File pull down menu, choose "Active Modules". Turn everything off "except" the following add-on modules:

Vega 3.X

DIS/HLA

Base3.X

### **Step 3: Create the battlefield.**

From the Icon Column, select the 'object' icon

From the object's pull down menu select 'New'

When prompted name the object 'town'

Assign :

NT:

c:\program files\paradigm\Vega\sample\data\Vega\testdb.flt as the geometry file

UNIX:

/usr/local/PSI/demo/testdb.flt as the geometry file

Assign isector bit mask for terrain mapping

On the Objects panel, select the button called "Object Isector Class".

After selecting the Override Option, set the Isector class with all "terrain bits" turned "on" ( terrain bits 1-8).

Assign DIS attributes to the town.

Under the Objects pull down menu, select the "Inbound Template Entity Information".

Disable all check boxes, since this object will be used only by you, (no PDU's need to be sent for the "terrain").

Close this dialogue box.

Under the Objects pull down menu select the "Outbound Template Information:

Disable all check boxes, since this object will not send any PDUs either.

Close this dialog box

Save your ADF as DISPlayer1.adf

### **Step 4: Configure your local entity:**

From the object's pull down menu select "New"

When prompted name the object "my tank"

Assign :

NT:

c:\program files\paradigm\Vega\sample\data\Vega\tank\_m1.flt as the geometry file

UNIX:

/usr/local/PSI/demo/tank\_m1.flt as the geometry file

Set the Coordinate System to "Dynamic"

Assign DIS attributes for your local entity

High light "my tank" in the entity list.

Under the Objects pull down menu, select the "Inbound Template Entity Information" Option.

Disable all check boxes, since this object will be used only a local entity.

Close this dialogue box.

*NOTE: On NT, the Inbound Template Entity Information dialog box is a "tab".*

*When you have made you selections, please select the tab labeled “Vega”.*

Highlight “my tank”

Under the Objects pull down menu, select the "Outbound Entity Information".

Check the “Enable for Sending” box

Set the Dead Reakoning to Rot/ConstPosRate/WorldCoords.

Under the Entity Type set the following:

Kind: Platform

Domain: Land

All other values leave at the Default, 0.

Close this dialogue box.

NOTE: On NT the Outbound Template Entity Information dialog box is a “tab”.

When you have made you selections, please select the tab labeled “Vega”.

Under the Objects pull down menu, select the "Outbound Entity Appearance".

Marking Text : Assign an intuitive name i.e. ‘player 1’

Close this dialogue box.

NOTE: On NT the Outbound Entity Appearance dialog box is a “tab”.

When you have made you selections, please select the tab labeled “Vega”.

Save Your ADF!

### **Step 5: Configure your inbound template:**

From the object’s pull down menu select “New”

When prompted name the object “inbound tank”

Assign :

NT:

c:\program files\paradigm\Vega\sample\data\Vega\tank\_m1.flt as the geometry file

UNIX:

/usr/local/PSI/demo/tank\_m1.flt as the geometry file

Set the Coordinate System to “Dynamic”.

Configure your inbound template for DIS

Highlight “inbound tank”

Under the objects pull down menu, select "Inbound Template Entity Information".

Select the box labeled : “Enable for Receiving”

Under the Entity Type set the following:

Kind: Platform

Domain: Land

All other values remain 0.

Close this dialogue box.

*(NT Users please see earlier note!)*

Under the Objects pull down menu, select "Outbound Entity Information". Disable all check boxes, since this object will be used for the "Inbound Template Entity" only. It is considered a "remote entity".

### **Step 6: Placing your objects in the Scene.**

> Go to the Scenes panel.

Click on the left arrow of the Objects List widget. This will list all instances of this class. Load the "town " and "my tank" into the scene by selecting the "Apply" button for the "terrain" and the "OK" button for the "tank". The "incoming\_tank"

IS NOT LOADED into the scene by you!

DIS will handle this with an incoming PDU.

### **Step 7: Create an Isector for clamping the motion model to the ground .**

Go to the Isector panel.

Create an isector called "driving".

Set the Z offset to be 0.1

Make the target the "Object" "town"

Make the method to type "tripod".

Click on the "Target Isector Class" button to set the Isector Class. Turn on all terrain bits, (they should already be on by default.)

Save your ADF.

### **Step 8: Using the Drive Motion Model.**

Go to the Motion Models panel.

Define your Initial/Reset position to be (1000, 1000, 0, 0, 0, 0)

Set the Motion Model Type. to "Drive"

Make the Wheel turning radius 2.5 for better steering of your tank.

Click on the left arrow to pop up the class instance dialogue box for Isectors.

Click on OK to attach the "driving" isector.

Save your ADF.

### **Step 9: Creating a tank player.**

Go to the Players panel.

Create a player called "tank driver"

Use the " my tank" for the players geometry.

Assign the tank object by clicking on the left arrow

An objects list widget will appear

Highlight 'my tank' and press OK

Set the "Positioning Method" to "Motion Model".

Enable the Motion model by checking the box labeled "Enable".

Select the "Default" Motion Model from the list below the Enable Widget.

This in essence has attached “my tank” to “tank driver”. “tank driver” is now tied to your motion model. Where ever the motion model goes, the tank will follow. Your motion model is control by your mouse.

Left mouse button will accelerate the drive motion model.

Right mouse button will slow down the drive motion model.

Middle mouse button will stop the drive motion model.

#### **Step 10: Have an Observer follow the tank driver.**

Go to the Observers panel.

Change the Default Observer’s “Positioning Method” from Type from "Motion Model" to "Tether Follow".

Attach the "tank driver" as the player the observer will follow.

Set the Offset to (0, -25, 5)

Save your ADF.

#### **Step 11: Configuring the machine for sending and receiving PDU’s.**

Go to the DIS panel.

Set the “Protocol” to “DIS”

Set the “Status” to “Send and Receive”

Host value is arbitrary. For this exercise use ‘111’

Each person must have a unique Host ID, in order to see everyone else in the application.

Turn the Display Entity Markings "ON" by clicking on the check box.

Attach the DIS Scene to the "Default" scene.

Save your ADF

Now preview your application using "DIS Active Preview".

You should see your tank. Your entity markings are displayed on the tank, (white is inbound and red outbound).

At this point there are no other participants! You must create a second ADF to participate in your DIS exercise.

### **Step 12: Create the player2.adf**

Copy the DISplayer1.adf to DISplayer2.adf

Go to the File pulldown menu.

Select "Save As" use "DISplayer2.adf" as the file name

To DISplayer2.adf Make the following modifications:

Go to the DIS panel.

Host : Because this is the second player, set the host value to "**222**"

Note: Each person must have a unique Host ID, in order to see everyone else in the application.

Save your ADF

Go to the "Object" Panel

Highlight "my tank" in the entity list.

Under the Objects pull down menu, select the "Outbound Entity Appearance".

Marking Text : Assign an intuitive name i.e. 'player 2'

Close this dialogue box.

Save your ADF. (DISplayer2.adf)

Go to the Motion Models panel.

Define your Initial/Reset position to be (1000, 1050, 0, 180, 0, 0)

Preview the application, again using "DIS Active Preview" for DISplayer1.adf and start a second DIS Active Preview Session for DISplayer2.adf.

You should be able to see both player1 and player 2 in both DIS Active Preview windows.



## Summary:

This tutorial will get you started setting up the basic communication between Vega DIS applications. For a more exciting demonstration of DIS please see

NT:

C:\program files\paradigm\vega\sample\vgDis\wargame.

Drag the wargame.adf over the wargame.exe.

Drag the wargame\_pl2.adf over the wargame.exe

IRIX

/usr/local/PSI/sample/wargame

type 'tank wargame.adf' on the command line

open a second window and type "tank wargame\_pl2.adf" on the command line

Supported keystrokes:

"l" – turn turret to the left

"r" – turn turret to the right

"u" – point barrel up

"d" – point barrel down

"F" fire a munitions

If you or the inbound player is hit by a missile, the tank will burst into flames! Your tank will also change from it's 'perfect' appearance into a 'damaged' appearance.

Question or problems please contact MPI Technical support  
e-mail [vega@multigen-paradigm.com](mailto:vega@multigen-paradigm.com)